

Django Form 품나게 쓰기

Django Form 원리부터 활용까지

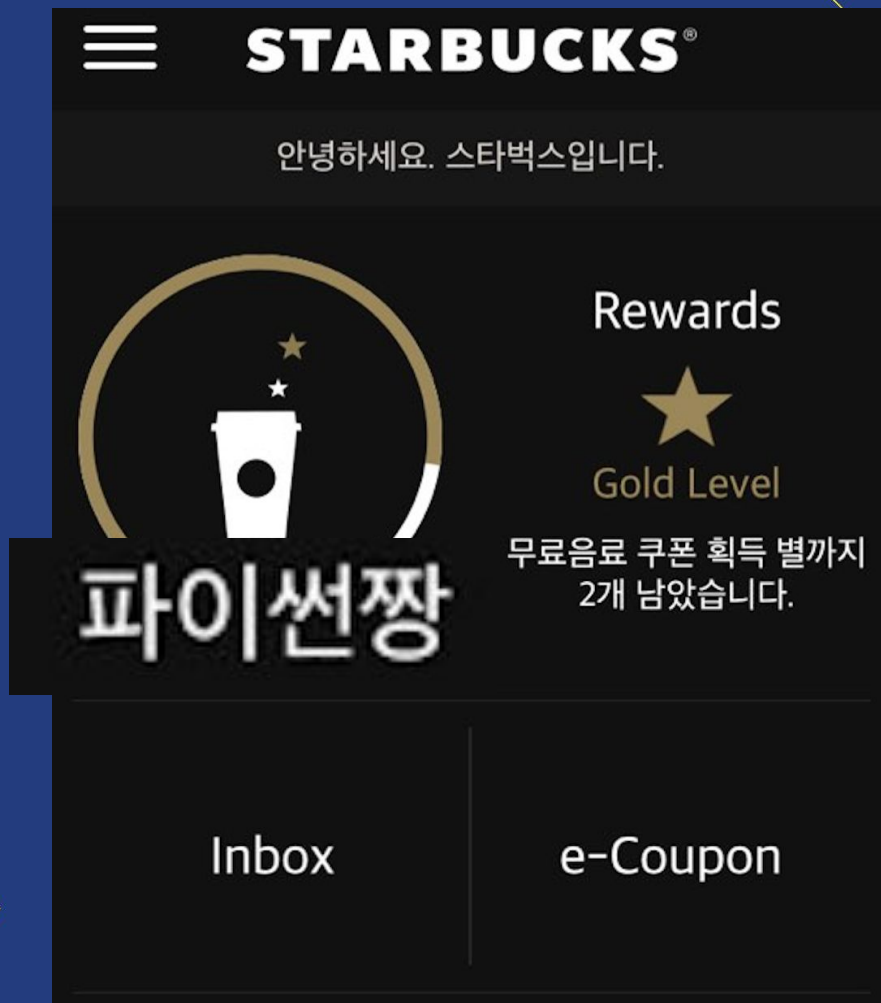
강승형

Back to the Basic

파이썬을 사랑하는 아재 개발자

발표자 소개

Back to the Basic



발표 소개

오늘의 이야기

Back to the Basic

Django Form?

Internal

ModelForm

FormView, GenericView

Back to the Basic

Django Form?

Back to the Basic

Web의 Form 처리 추상화

Back to the Basic

Name***Desc**

Django를 Full stack으로 사용하면서 가장 많이 고민해던 Form에 대한 부분의 경험을 공유하기 위한 발표입니다. 기본적으로 Django 내부의 Form의 동작원리를 살펴보고, Django에서 권장하는 Djangoic한 방법의 Form 사용법도 살펴보고, 실제 로직 구현시 부딪히게된 문제와 해결 방법을 공유할 예정입니다.

Language***Back to the Basic**

성공 : Redirect

Back to the Basic

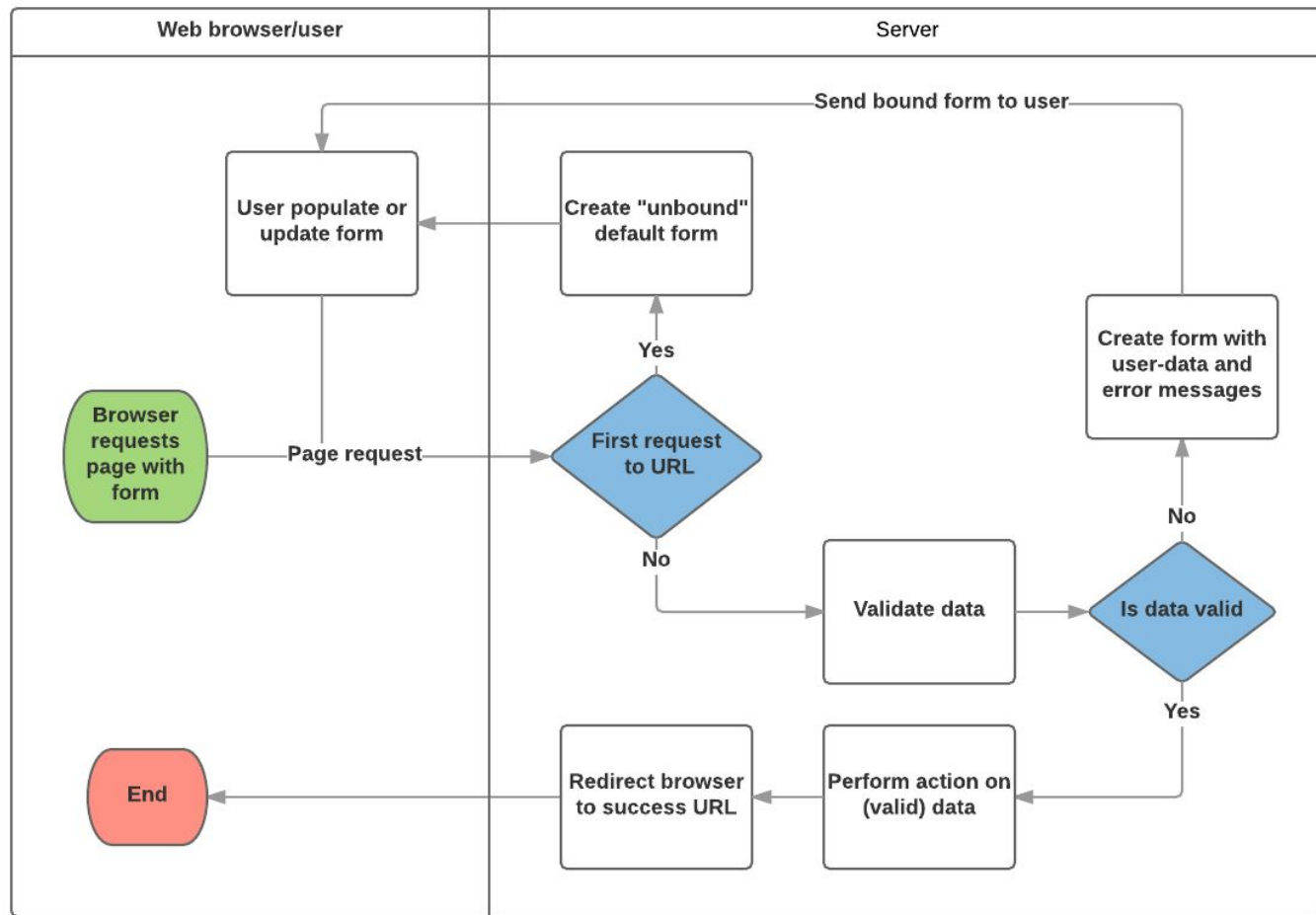
실패 : Error

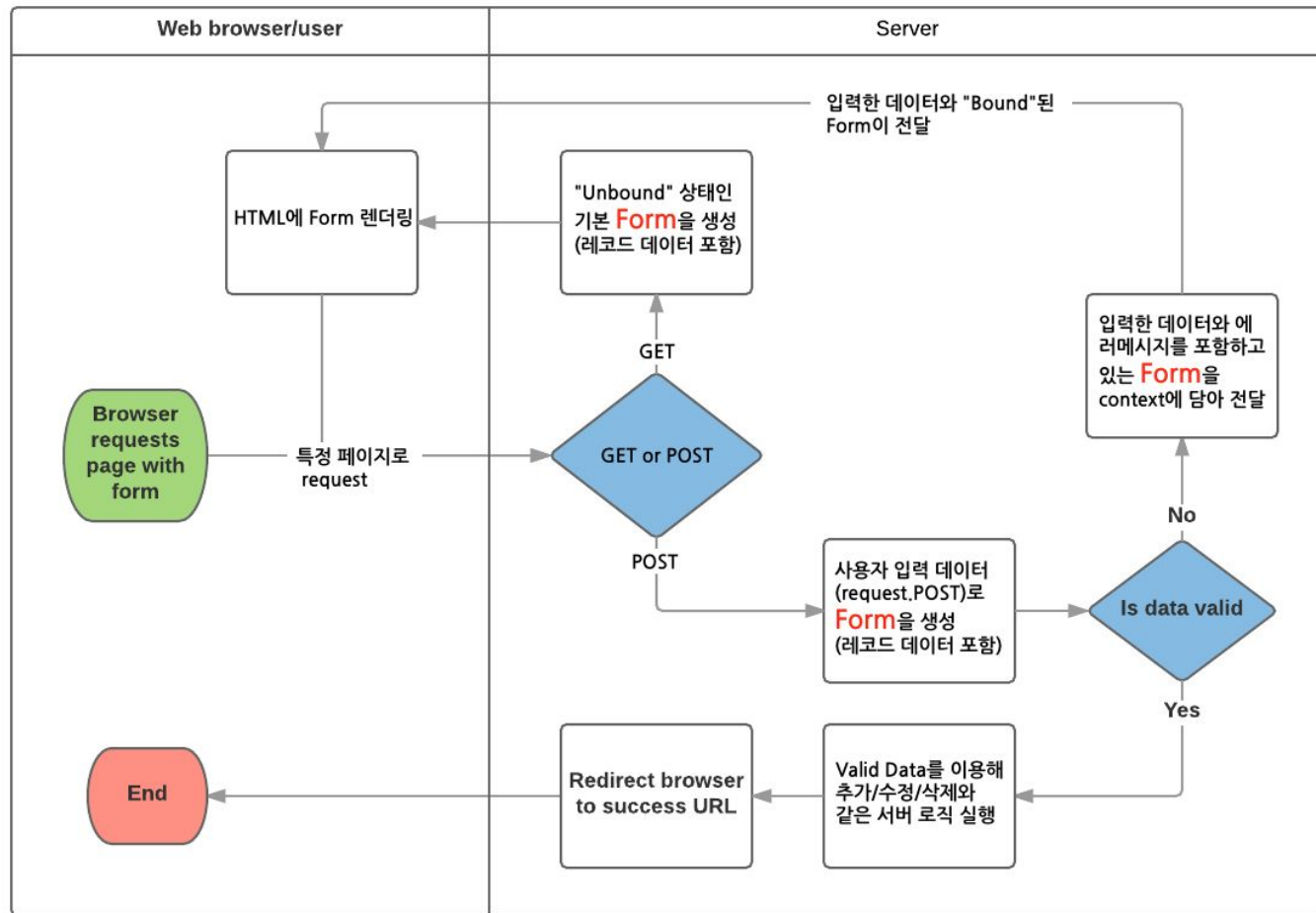
Back to the Basic

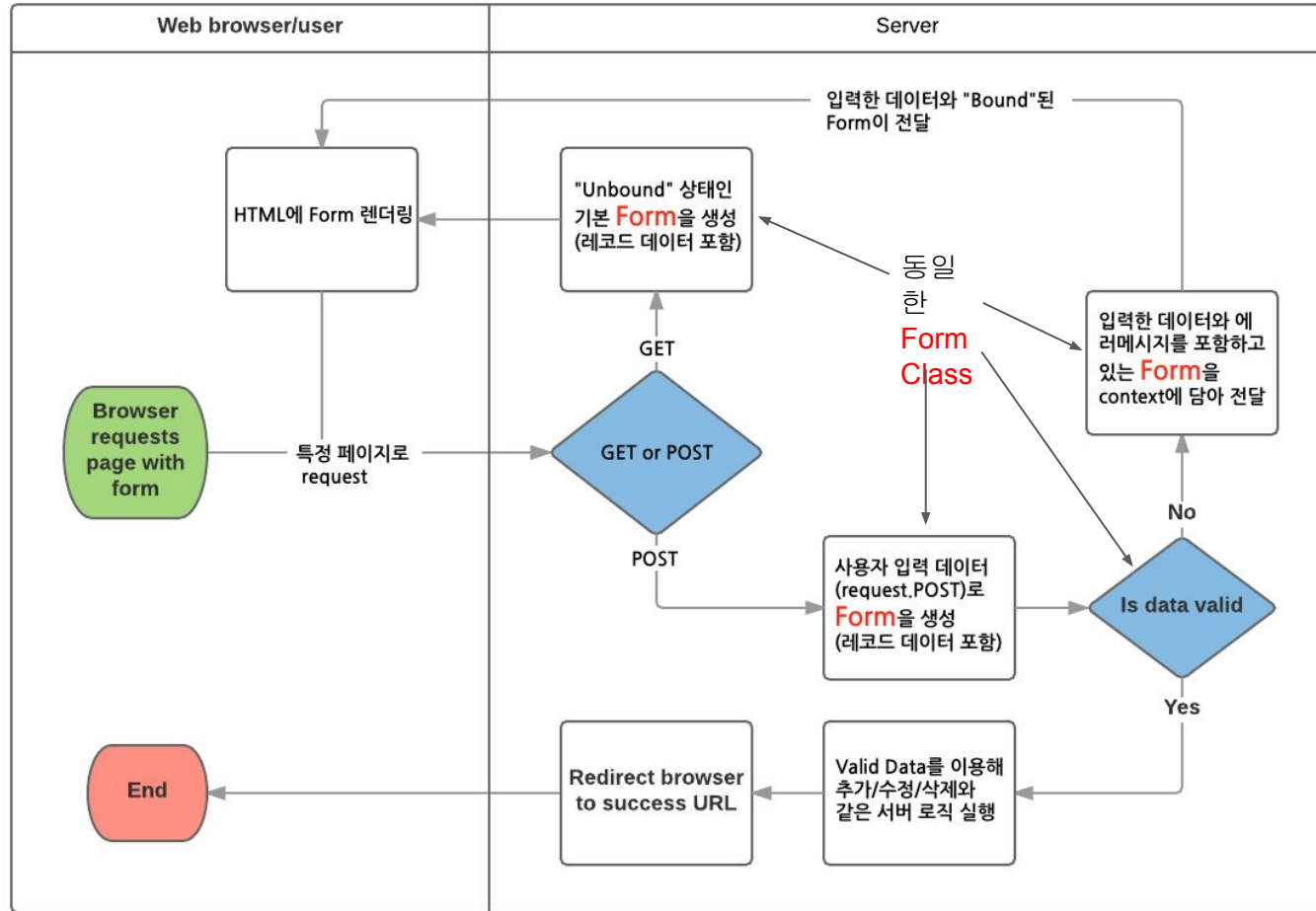
Name***필수 항목입니다.****Desc**

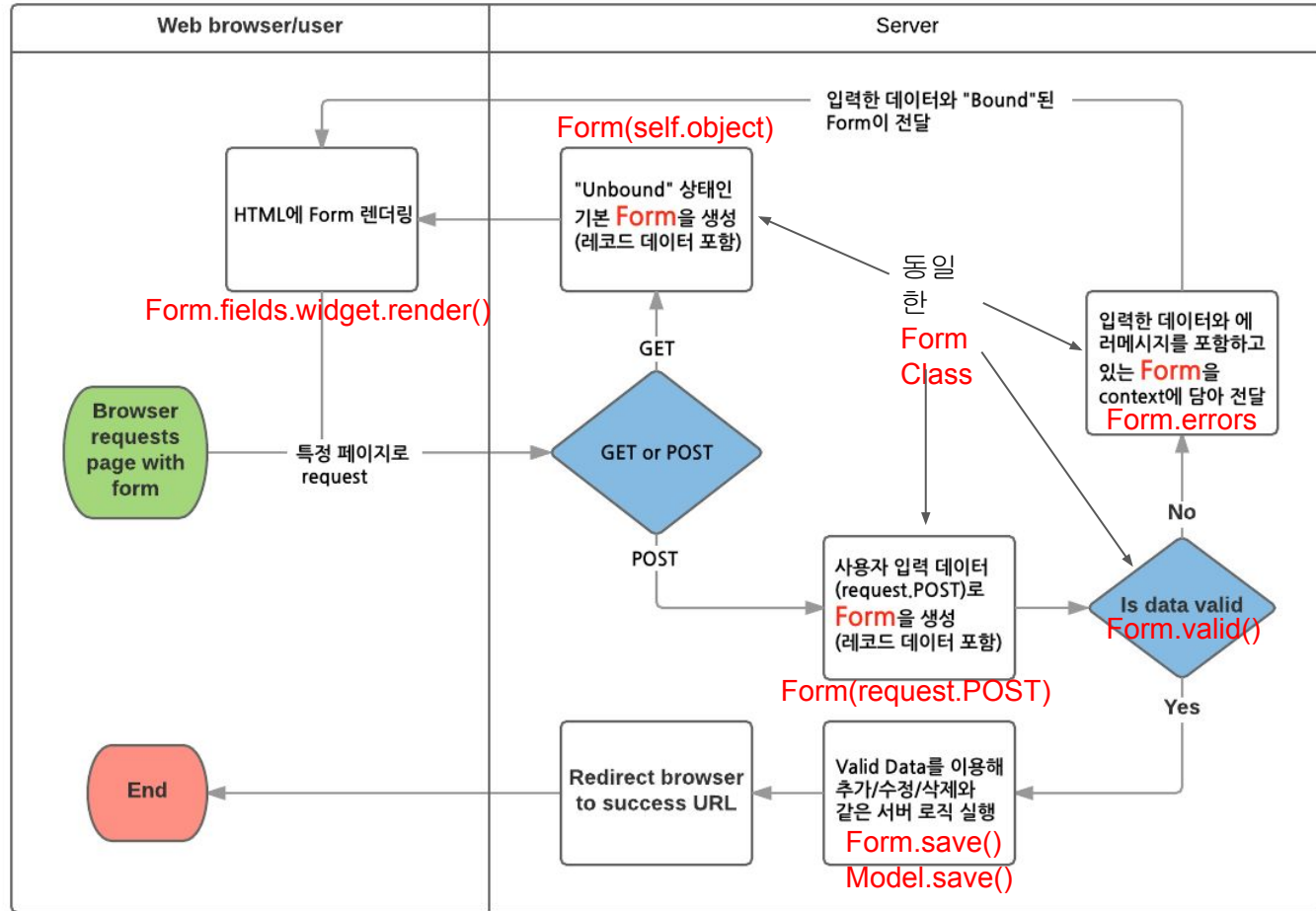
Django를 Full stack으로 사용하면서 가장 많이 고민해던 Form에 대한 부분의 경험을 공유하기 위한 발표입니다. 기본적으로 Django 내부의 Form의 동작원리를 살펴보고, Django에서 권장하는 Djangonic한 방법의 Form 사용법도 살펴보고, 실제 로직 구현시 부담히게된 문제와 해결 방법을 공유할 예정입니다.

Language***전송****Back to the Basic**









Django Form Class

Back to the Basic


```
class Form:
```

```
    fields = []
```

```
    data = {}
```

```
    initial = {}
```

```
    cleaned_data, errors = {}, []
```

```
    def __init__(self, data=None):
```

```
        self.is_bound = data is not None
```

```
    def is_valid(self):
```

```
        self.clean()
```

```
        return self.is_bound and not self.errors
```

```
    def clean(self):
```

```
        for name, field in self.fields:
```

```
            try:
```

```
                value = field.clean()
```

```
            except ValidationError as error:
```

```
                self.errors.append(error)
```

```
            else:
```

```
                self.cleaned_data[name] = value
```

class Field:

widget = TextInput

validators = []

def __init__(self, validators):

self.validators = validators

def clean(self, value):

self.run_validators(value)

return value

def run_validators(self, value):

errors = []

for validator in self.validators:

try:

validator(value)

except ValidationError as error:

errors.append(error)

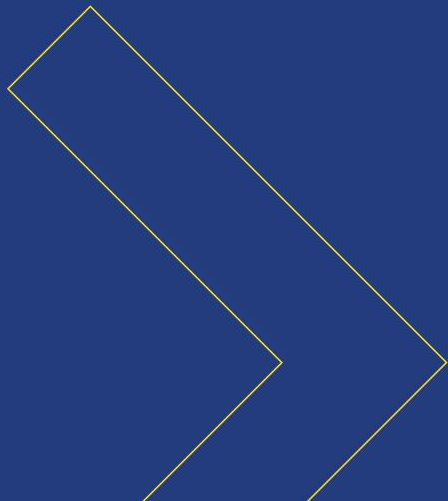
if errors:

raise ValidationError(errors)

```
class MyForm(forms.Form):  
    field1 = forms.IntegerField(min_value=0, max_value=100000, required=True)  
    field2 = forms.CharField(max_length=10, validators=[regex], required=False)  
    field3 = forms.BooleanField(widget=forms.CheckboxInput, required=True)  
  
    def clean_field1(self):  
        cleaned_field1 = self.cleaned_data['field1']  
        if not self.additional_validate(cleaned_field1):  
            raise forms.ValidationError(  
                self.error_messages['field1_incorrect'],  
                code='field1_incorrect',  
            )  
        return cleaned_field1
```



Model Form?



Back to the Basic

ModelForm = Form from Model

Back to the Basic

Form \approx Model

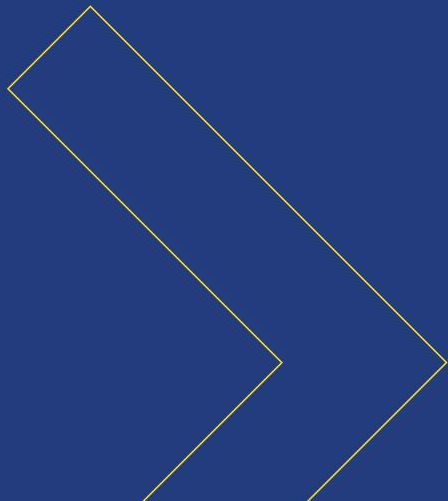
Back to the Basic

Form.field \approx Model.field

Back to the Basic



`Form.save() ≈ Model.save()`



Back to the Basic

Django Form 폼나게 쓰기

제목*

Django Form 폼나게 쓰기

슬라이드 URL

동영상 URL

☒ 촬영, 녹화가 가능합니다.

설명

 **B** *I* U ~~X'~~ ~~X,~~   16▼ **A** ▼    ▼ **T!** ▼  ▼       

Django를 Full stack으로 사용하면서 가장 많이 고민해던 Form에 대한 부분의 경험을 공유하기 위한 발표입니다. 기본적으로 Django 내부의 Form의 동작원리를 살펴보고, Django에서 권장하는 Djangonic한 방법의 Form 사용법도 살펴보고, 실제 로직 구현시 부딪히게된 문제와 해결 방법을 공유할 예정입니다.

간단하게 Modeling

```
class Program(models.Model):  
    name = models.CharField(max_length=255, db_index=True)  
    desc = models.TextField(null=True, blank=True)  
    language = models.CharField(max_length=1, choices=(  
        ('E', _('English')), ('K', _('Korean')), ), default='E')
```

```
class Url(models.Model):  
    program = models.ForeignKey(Program)  
    category = models.CharField(max_length=1, choices=(  
        ('S', _('Slide')), ('P', _('PDF')), ('V', _('Video')), ) )  
    location = models.CharField(max_length=255)  
    ORDER = models.IntegerField(default=0, null=True)
```

ModelForm은 초간단

```
class ProgramForm(ModelForm):  
    class Meta:  
        model = Program  
        fields = ['name', 'desc', 'language']
```

```
class UrlForm(ModelForm):  
    class Meta:  
        model = Url  
        fields = '__all__'
```

Template도 초간단

```
<form method="post">
```

```
{% csrf_token %}
```

```
{{ form.as_p }}
```

```
<input type="submit" value="전송"/>
```

```
</form>
```

Name:

Django Form 폼나게 쓰기

Desc:

Language:

Back to the Basic

View는 복잡

```
class ProgramCreate(TemplateView):
```

```
    form_class = ProgramForm
```

```
    template_name = 'program_form.html'
```

```
    success_url = reverse_lazy('program:list')
```

```
    def post(self, request, *args, **kwargs):
```

```
        form = self.form_class(data=request.POST, instance=self.object)
```

```
        if form.is_valid():
```

```
            form.save()
```

```
            return HttpResponseRedirect(self.success_url)
```

```
        else:
```

```
            return self.render_to_response(self.get_context_data(form=form))
```

```
    def get_context_data(self, **kwargs):
```

```
        if 'object' not in kwargs:
```

```
            kwargs['object'] = self.object
```

```
        if 'form' not in kwargs:
```

```
            kwargs['form'] = self.form_class(instance=self.object)
```

```
        return super().get_context_data(**kwargs)
```

FormView로 간단히

```
class ProgramFormViewCreate(ModelFormMixin, FormView):  
    form_class = ProgramForm  
    object = None  
    template_name = 'program_form_formview.html'  
    success_url = reverse_lazy('program_formview:list')
```

GenericView로 초간단

```
from django.views.generic import CreateView  
class ProgramGenericCreateView(CreateView):  
    model = Program  
    fields = ['name', 'desc', 'language']  
    success_url = reverse_lazy('program_generic:list')
```

세상에! Form도 작성할 필요없어요!

Formsets

Back to the Basic

Formsets = a Set of Form

Back to the Basic

Program

Name:

Django Form 폼나게 쓰기!

Desc:

Language:

URL

Category: Location: 순서: 삭제: ☐

Category: Location: 순서: 삭제: ☐

Category: Location: 순서: 삭제: ☐

django-crispy-forms

Back to the Basic

Name: Django Form 폼나게 쓰기!

Django를 Full stack으로 사용하면서 가장 많이 고민
Form에 대한 부분의 경험을 공유하기 위한 발표입니다.
기본적으로 Django 내부의 Form의 동작원리를 살펴보고,
Django에서 권장하는 Django-like한 방법의 Form 사
도 살펴보고, 실제 로직 구현시 부딪히게된 문제와 해결
법을 공유할 예정입니다.

Before

Desc:

Language: 영어

전송

Name*

Django Form 폼나게 쓰기!

Desc

Django를 Full stack으로 사용하면서 가장 많이 고민했던 Form에 대한 부
분의 경험을 공유하기 위한 발표입니다. 기본적으로 Django 내부의 Form
의 동작원리를 살펴보고, Django에서 권장하는 Django-like한 방법의
Form 사용법도 살펴보고, 실제 로직 구현시 부딪히게된 문제와 해결 방법을
공유할 예정입니다.

After

Language*

영어

전송

감사합니다.

Back to the Basic